

ПОБУДОВА АРХІТЕКТУРИ MVC ДЛЯ ВЕБДОДАТКІВ НА ОСНОВІ SPRING FRAMEWORK

Вінницький національний технічний університет

Анотація

У статті розглядається архітектурний патерн Model-View-Controller (MVC) для створення вебдодатків. Детально описано розподіл відповідальностей між компонентами Model, View та Controller. Представлено реалізацію MVC у Spring Framework, включаючи налаштування контролерів, сервісів і шаблонів представлення. Також розглянуто переваги використання цього підходу для побудови масштабованих та підтримуваних вебзастосунків.

Ключові слова: MVC, Spring, веб-додаток, Model, View, Controller, Java.

Abstract

The article discusses the Model-View-Controller (MVC) architectural pattern for creating web applications. The distribution of responsibilities between Model, View, and Controller components is described in detail. The MVC implementation in the Spring Framework is presented, including the configuration of controllers, services, and view templates. The advantages of using this approach to build scalable and maintainable web applications are also discussed

Keywords: MVC, Spring, web app, Model, View, Controller, Java.

Вступ

Патерн MVC є основою для багатьох сучасних вебфреймворків, зокрема Spring MVC. Він забезпечує поділ логіки додатка на три основні компоненти:

1. Model (Модель) – відповідає за управління даними, бізнес-логікою та взаємодію з базою даних.
2. View (Представлення) – відповідає за відображення інформації користувачу.
3. Controller (Контролер) – приймає запити від користувача, викликає відповідні сервіси та повертає відповідь.

У даній роботі розглянуто реалізацію MVC у Spring Framework та переваги його використання.

Результати дослідження

Алгоритм побудови вебдодатка на основі MVC включає кілька етапів:

1. Налаштування середовища.
Використовується Spring Boot, який забезпечує швидке налаштування та інтеграцію MVC.
2. Створення моделі.
Визначаються сутності (Entity) та сервіси (Service), які містять бізнес-логіку.
3. Розробка контролера.

Контролер (Controller) обробляє HTTP-запити, викликає необхідні сервіси та передає дані у представлення.

4. Створення представлення.
Використовуються шаблони, такі як Thymeleaf або JSP, для відображення даних користувачу.
5. Запуск і тестування.

Додаток розгортається на сервері Tomcat, де перевіряється його коректна робота.

На рисунку 1 зображено класичну схему MVC.

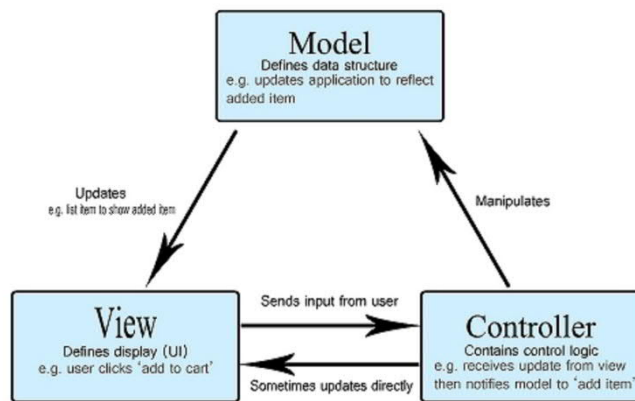


Рисунок 1 – Класична схема MVC

На рисунку представлено загальну схему роботи патерну MVC, де Model відповідає за управління даними, View – за їх відображення, а Controller – за обробку запитів користувачів та взаємодію між компонентами. Такий підхід забезпечує розділення логіки додатка, що сприяє його масштабованості та зручності в супроводі.

Висновки

У роботі було розглянуто архітектурний патерн MVC та його реалізацію у Spring Framework. Було визначено, що MVC сприяє чіткому розподілу логіки додатка та спрощує підтримку коду. Використання Spring MVC дозволяє швидко розгорнути веб-додатки з мінімальною конфігурацією. Підхід MVC може бути адаптований для мікросервісної архітектури та SPA-додатків.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Craig Walls. Spring in Action. 6th Edition, Manning, 2022.
2. Martin Fowler. Patterns of Enterprise Application Architecture. Addison-Wesley, 2002.
3. Офіційна документація Spring: <https://spring.io/projects/spring-framework>

Артем Юрійович Білінський – студент ЗПІ-23Б факультету інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: bilinskiyartem110407@gmail.com;

Artem Y. Bilinskyi – Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: bilinskiyartem110407@gmail.com;